

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Hiroki HASEGAWA et al.

Application No.:

Group Art Unit:

Filed: March 22, 2004

Examiner:

For: APPARATUS FOR ADJUSTING USE RESOURCES OF SYSTEM AND METHOD
THEREOF

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s) herewith
a certified copy of the following foreign application:

Japanese Patent Application No(s). 2003-111301

Filed: April 16, 2003

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing
date(s) as evidenced by the certified papers attached hereto, in accordance with the
requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: 3/23/04

By: Richard A. Gollhofer
Richard A. Gollhofer
Registration No. 31,106

1201 New York Ave, N.W., Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501

JAPAN PATENT OFFICE

This is to certify that the annexed is a true copy of the following application as filed with this Office.

Date of Application: April 16, 2003

Application Number: Patent Application
No. 2003-111301
[ST.10/C]: [JP2003-111301]

Applicant(s): FUJITSU LIMITED

December 9, 2003

Commissioner,
Japan Patent Office Yasuo IMAI

Certificate No. P2003-3101652

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 2003年 4月16日
Date of Application:

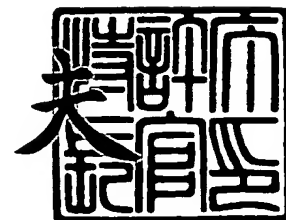
出願番号 特願2003-111301
Application Number:
[ST. 10/C]: [JP 2003-111301]

出願人 富士通株式会社
Applicant(s):

2003年12月 9日

特許庁長官
Commissioner,
Japan Patent Office

今井 康



出証番号 出証特2003-3101652

【書類名】 特許願

【整理番号】 0350863

【提出日】 平成15年 4月16日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 9/50

【発明の名称】 システムの使用資源を調整する装置および方法

【請求項の数】 9

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社内

【氏名】 長谷川 博己

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社内

【氏名】 葉袋 克仁

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社内

【氏名】 植松 俊晃

【特許出願人】

【識別番号】 000005223

【氏名又は名称】 富士通株式会社

【代理人】

【識別番号】 100074099

【住所又は居所】 東京都千代田区二番町 8 番地 2 0 二番町ビル 3 F

【弁理士】

【氏名又は名称】 大菅 義之

【電話番号】 03-3238-0031

【選任した代理人】

【識別番号】 100067987

【住所又は居所】 神奈川県横浜市鶴見区北寺尾 7-25-28-503

【弁理士】

【氏名又は名称】 久木元 彰

【電話番号】 045-573-3683

【手数料の表示】

【予納台帳番号】 012542

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9705047

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 システムの使用資源を調整する装置および方法

【特許請求の範囲】

【請求項 1】 1つ以上のアプリケーションプログラムからなるモジュールを複数有するシステムにおいて使用されるコンピュータ資源の量をモジュール毎に調整する資源調整装置であって、

複数のモジュールのそれぞれについて、過去のトランザクション発生量の推移を表すデータを格納する格納手段と、

前記格納手段から対象モジュールのトランザクション発生量の推移を表すデータを取得し、該対象モジュールの過去のトランザクション処理量と使用資源量の相関関係を表す関数におけるトランザクション処理量としてトランザクション発生量を用いることで、該対象モジュールのトランザクション発生量の推移から使用資源量の推移を生成する生成手段と、

生成された使用資源量の推移を予測される使用資源量の推移として用い、該予測される使用資源量の推移に従って、前記対象モジュールの割当資源量を自動的に変動させる割当手段と

を備えることを特徴とする資源調整装置。

【請求項 2】 1つ以上のアプリケーションプログラムからなるモジュールを複数有するシステムにおいて使用されるコンピュータ資源の量をモジュール毎に調整するコンピュータのためのプログラムであって、

複数のモジュールのそれぞれについて過去のトランザクション発生量の推移を表すデータを格納する格納手段から、対象モジュールのトランザクション発生量の推移を表すデータを取得し、

前記対象モジュールの過去のトランザクション処理量と使用資源量の相関関係を表す関数におけるトランザクション処理量としてトランザクション発生量を用いることで、前記対象モジュールのトランザクション発生量の推移から使用資源量の推移を生成し、

生成された使用資源量の推移を予測される使用資源量の推移として用い、該予測される使用資源量の推移に従って、前記対象モジュールの割当資源量を自動的に

に変動させる

処理を前記コンピュータに実行させることを特徴とするプログラム。

【請求項 3】 前記プログラムは、前記対象モジュールのトランザクション発生量の推移を表すデータから、複数種類のサイクルのそれぞれにおけるトランザクション発生量の推移を生成して画面上に表示し、オペレータからの指示に従って各サイクルにおけるトランザクション発生量の推移を合成することで、予測されるトランザクション発生量の推移を生成し、該予測されるトランザクション発生量の推移に前記関数を適用して前記使用資源量の推移を生成する処理を前記コンピュータに実行させることを特徴とする請求項 2 記載のプログラム。

【請求項 4】 前記プログラムは、前記複数種類のサイクルのそれぞれにおいて、前記システム内の 2 つ以上のモジュールに関するトランザクション発生量の平均値および最大値の推移を生成して前記画面上に表示し、前記オペレータが選択した値を用いて各サイクルにおけるトランザクション発生量の推移を合成することで、前記予測されるトランザクション発生量の推移を生成する処理を前記コンピュータに実行させることを特徴とする請求項 3 記載のプログラム。

【請求項 5】 前記プログラムは、前記生成された使用資源量の推移を画面上に表示し、オペレータが表示された使用資源量の推移を変更したとき、変更された使用資源量の推移を前記予測される使用資源量の推移として用いる処理を前記コンピュータに実行させることを特徴とする請求項 2 記載のプログラム。

【請求項 6】 前記プログラムは、前記対象モジュールの使用資源量があらかじめ決められたボトルネック検出閾値に達したときから、該対象モジュールの使用資源量があらかじめ決められたボトルネック解消閾値に達するまでの間、該対象モジュールに優先的に資源を割り当てる処理を前記コンピュータに実行させることを特徴とする請求項 2 記載のプログラム。

【請求項 7】 前記プログラムは、前記対象モジュールのトランザクション発生量があらかじめ決められたボトルネック検出閾値に達したときから、該対象モジュールのトランザクション発生量があらかじめ決められたボトルネック解消閾値に達するまでの間、該対象モジュールに優先的に資源を割り当てる処理を前記コンピュータに実行させることを特徴とする請求項 2 記載のプログラム。

【請求項 8】 前記プログラムは、前記対象モジュールの予測される使用資源量があらかじめ決められた値に達したとき、該対象モジュールに対して子プロセスの生成を指示する処理を前記コンピュータに実行させることを特徴とする請求項 2 記載のプログラム。

【請求項 9】 1 つ以上のアプリケーションプログラムからなるモジュールを複数有するシステムにおいて使用されるコンピュータ資源の量をモジュール毎に調整する資源調整方法であって、

生成手段が、複数のモジュールのそれぞれについて過去のトランザクション発生量の推移を表すデータを格納する格納手段から、対象モジュールのトランザクション発生量の推移を表すデータを取得し、

前記生成手段が、前記対象モジュールの過去のトランザクション処理量と使用資源量の相関関係を表す関数におけるトランザクション処理量としてトランザクション発生量を用いることで、前記対象モジュールのトランザクション発生量の推移から使用資源量の推移を生成し、

割当手段が、生成された使用資源量の推移を予測される使用資源量の推移として用い、該予測される使用資源量の推移に従って、前記対象モジュールの割当資源量を自動的に変動させる

ことを特徴とする資源調整方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、情報処理システム各モジュールが将来において使用するコンピュータ資源の量を予測し、モジュール毎の使用資源量の制限を自律的に調整する装置および方法に関する。

【0002】

【従来の技術】

図 17 は、従来の情報処理システムにおけるソフトウェアモジュールを示している。システム内にあるアプリケーションプログラム（業務）の集まりがモジュールであり、1 つのモジュールは 1 つ以上のアプリケーションからなる。

【0003】

図17では、モジュール1は、業務1、業務2、および業務3の3つのアプリケーションからなり、モジュール2は、2つの業務のアプリケーションからなり、モジュール3は、1つの業務のアプリケーションからなっている。

【0004】

このようなシステムにおいては、モジュール毎に使用資源量の制限を設けて資源量制御を行うことで、アプリケーションが使用する資源の消耗を回避し、可用性のある予測可能なレベルのサービスを実現することができる（例えば、非特許文献1、特許文献1、2、3、および4参照）。

【0005】

【非特許文献1】

“Solaris 9 Resource Manager”、[online]、Sun Microsystems—製品 & サービス—Sun ONE ソフトウェア—Solaris—Solaris 9—データシート、[平成15年3月11日検索]、インターネット<URL : <http://jp.sun.com/software/solaris/ds/ds-srm/>>

【特許文献1】

特開2002-268920号公報

【特許文献2】

特開2002-342269号公報

【特許文献3】

特開2001-022597号公報

【特許文献4】

特開平10-124333号公報

【0006】

【発明が解決しようとする課題】

しかしながら、上述した従来の資源量制御には、次のような問題がある。

従来の資源量制御では、モジュールをアプリケーションと関連付け、モジュールの使用資源量に制限を設定している。これにより、使用資源量が大きいアプリケーションには大きな割当資源量を設定し、使用資源量が小さいアプリケーション

ンには小さな割当資源量を設定して、アプリケーションが使用する資源の消耗を回避することができる。

【0007】

しかし、各アプリケーションの使用資源量は一定ではなく、分・時・週等により変化する。また、突発的に使用資源量が増加し、ボトルネックに陥ることもある。このような使用資源量の変化に対応するためには、システム管理者が使用資源量を随時監視し、状況に応じて制限の設定を一々変更しなければならない。

【0008】

また、使用資源量はトランザクション処理量に比例し、必ずしもトランザクション発生量に比例するわけではないので、単に過去の使用資源量の推移を参照するだけでは、今後のトランザクション発生量に対する使用資源量を予測することはできない。

【0009】

本発明の課題は、情報処理システムにおける各モジュールの使用資源量の変化を的確に予測し、その変化に応じて使用資源量の制限を自律的に調整する装置および方法を提供することである。

【0010】

【課題を解決するための手段】

図1は、本発明の資源調整装置の原理図である。図1の資源調整装置は、格納手段101、生成手段102、および割当手段103を備え、1つ以上のアプリケーションプログラムからなるモジュールを複数有するシステムにおいて使用されるコンピュータ資源の量をモジュール毎に調整する。

【0011】

格納手段101は、複数のモジュールのそれぞれについて、過去のトランザクション発生量の推移を表すデータを格納する。生成手段102は、格納手段101から対象モジュールのトランザクション発生量の推移を表すデータを取得し、対象モジュールの過去のトランザクション処理量と使用資源量の相関関係を表す関数におけるトランザクション処理量としてトランザクション発生量を用いることで、対象モジュールのトランザクション発生量の推移から使用資源量の推移を

生成する。割当手段103は、生成された使用資源量の推移を予測される使用資源量の推移として用い、予測される使用資源量の推移に従って、対象モジュールの割当資源量を自動的に変動させる。

【0012】

格納手段101には、例えば、モジュール毎に収集されたトランザクション発生量の測定値が格納されており、生成手段102は、調整対象となるモジュール（対象モジュール）のトランザクション発生量の測定値を格納手段101から取得する。次に、生成手段102は、対象モジュールの過去のトランザクション処理量と使用資源量の相関関係を表す関数に、トランザクション発生量の測定値をトランザクション処理量として代入することで、対象モジュールの使用資源量の推移を算出する。そして、割当手段103は、算出された使用資源量の推移に合わせて、今後の対象モジュールの割当資源量を自動的に変動させる。

【0013】

このように、過去のトランザクション発生量をトランザクション処理量に当てはめて今後の使用資源量を算出することで、モジュール毎の使用資源量の変化を的確に予測することができる。また、予測された使用資源量の推移に従って割当資源量を変動させることで、モジュール毎の使用資源量の制限を自律的に変動させることができる。

【0014】

格納手段101は、例えば、後述する図15の外部記憶装置1505に対応し、生成手段102および割当手段103は、例えば、図15のCPU（中央処理装置）1501およびメモリ1502の組み合わせに対応する。

【0015】

【発明の実施の形態】

以下、図面を参照しながら、本発明の実施の形態を詳細に説明する。

本実施形態では、コンピュータ資源を管理することができる情報処理システムにおいて、モジュール毎に使用資源量の制限に関するスケジューリングを行い、資源の有効利用、ボトルネックの回避、システム全体のキャパシティプランニング支援等を実現する。

【0016】

資源調整装置は、各アプリケーションに関連付けたモジュール毎にトランザクション量（発生量・処理量）と使用資源量の測定値を蓄積し、トランザクション発生量の各サイクル（例えば、日、週、月）のパターン（平均値もしくは最大値）を求める。そして、得られた各サイクルのパターンを合成して、特定のサイクル単位（例えば、月単位）のトランザクション発生量の推移を算出する。

【0017】

また、資源調整装置は、トランザクション処理量と使用資源量の関係を表す近似関数を求め、この近似関数を月単位のトランザクション発生量の推移に適用して、月単位の予測される使用資源量の推移を求める。ここでの近似関数は、通常用いられる実数の近似を一般化したものと考えればよい。すなわち、実数の近似値がある数値的な量を特定の桁精度で表すものとすれば、近似関数は単一もしくは複数のパラメータに依存した量を特定の精度で表すものとする。

【0018】

このようにして、今後の使用資源量をスケジューリングすることで、モジュール毎の使用資源量の制限を自動的かつ的確に変動させることができる。

また、資源調整装置は、各モジュールにボトルネック検出閾値を設定しておき、使用資源量がその閾値に達した場合に、そのモジュールに優先的に資源を割り当てる。これにより、ボトルネックを解消することができる。

【0019】

資源調整装置の運用モードは、監視運用モードと自律運用モードの2つのモードに大別され、資源調整装置は、モジュール毎に必要な資源量をスケジューリングしてサーバの運用設計を行う。

【0020】

監視運用モードでは、スケジューリングされた資源量を管理者（オペレータ）が変更することが可能であり、資源調整装置は、最終的に管理者により決定されたスケジュールに従って資源量を調整する。また、自律運用モードでは、モジュール単位でトランザクション量と使用資源量に関する統計情報を収集し、モジュールの状態を自動的に判断して、モジュール毎の資源配分を調整する。

【0021】

図2は、監視運用モードの処理のフローチャートである。資源調整装置は、まず、モジュール単位でトランザクション量（発生量・処理量）と使用資源量に関する統計情報を収集する（ステップ201）。統計情報は、定期的に収集されて蓄積される。

【0022】

次に、収集された統計情報からモジュール単位の割当資源量（必要資源量）を算出して、資源量のスケジュールを管理者に提示する（ステップ202）。

ここでは、まず、過去に蓄積されたトランザクション発生量から、各サイクル（日、週、月等）毎のトランザクション発生量の推移と、システム内の複数のモジュールに関するトランザクション発生量の平均値および最大値の推移を求め、管理者に提示する。管理者は、各サイクル毎に平均値または最大値を選択することで、トランザクションの推移の合成を指示する。

【0023】

このとき、資源調整装置は、例えば、図3のような画面をディスプレイ上に表示する。図3の画面では、サイクルとして1日が選択されており、統計処理の期間として3日が選択されている。

【0024】

折れ線グラフ301、302、および303は、それぞれ12月1日、2日、および3日のあるモジュールにおけるトランザクション発生量の推移を表し、折れ線グラフ304は、この3日間におけるトランザクション発生量の平均値の推移を表し、折れ線グラフ305は、この3日間におけるトランザクション発生量の最大値の推移を表している。管理者は、マウス等のポインティングデバイスを用いて、表示された平均値または最大値（MAX値）を選択し、操作が終了すればOKボタン306をクリックする。

【0025】

同様にして、管理者は、他のサイクルについても平均値または最大値を選択し、資源調整装置は、選択された各サイクルの推移（パターン）を重ね合わせることで、今後の予測されるトランザクション発生量の推移を合成・算出する。

【0026】

これにより、例えば、図4のような画面が表示される。図4の画面では、“日”および“週”の各サイクルのパターンを加味した今後の“月”に関するトランザクション発生量の推移が算出されている。

【0027】

次に、資源調整装置は、過去のトランザクション処理量と使用資源量の相関関係を表す近似関数を生成する。例えば、近似関数として指数関数を用いた場合、A、B、およびCを定数として次のような近似式が生成される。

$$(\text{使用資源量}) = A * (\text{トランザクション処理量})^B + C$$

これにより、例えば、図5のような画面が表示される。図5の画面の横軸はトランザクション処理量を表し、縦軸は使用資源量を表している。

【0028】

次に、この近似関数と今後のトランザクション発生量の推移から、今後の使用資源量のスケジュールを算出して、管理者に提示する。これにより、例えば、図6のような画面が表示される。図6の画面では、今後の“月”に関する使用資源量のスケジュールとして、折れ線グラフ601、602、および603が表示されている。折れ線グラフ601、602、および603は、それぞれモジュール#1、#2、および#3のスケジュールを表す。

【0029】

管理者は、こうして提示された使用資源量のスケジュールの設定を、必要に応じて変更することができる（ステップ203）。管理者は、例えば、画面上でポインティングデバイスを用いて折れ線グラフを上下にドラッグすることにより、スケジュールを変更する（ステップ204）。そして、操作が終了すればOKボタン604をクリックする。

【0030】

資源調整装置は、スケジュールが変更されなかった場合は算出されたスケジュールに従って、スケジュールが変更された場合は変更後のスケジュールに従って

、今後の割当資源量を自動的に変動させる（ステップ205）。

【0031】

そして、その後の資源の割当状況を監視し、予測・設定された割当資源量と実際に使用された資源量とを比較して、比較結果を管理者に提示する（ステップ206）。管理者は、割当資源量と使用資源量が運用システムとして適合している場合には、監視運用モードから自律運用モードにモード変更を行うことができる。

【0032】

図7は、自律運用モードの処理のフローチャートである。図7のステップ701の処理は、図2のステップ201の処理と同様である。資源調整装置は、収集された統計情報からモジュール単位の割当資源量を算出する（ステップ702）。

【0033】

ここでは、まず、過去に蓄積されたトランザクション発生量から、次の時間単位（例えば、初期値は1時間）におけるトランザクション発生量を予測する。次に、監視運用モードの場合と同様に、過去のトランザクション処理量と使用資源量の相関関係を表す近似関数を生成する。そして、予測されたトランザクション発生量と近似関数から、次の時間単位で必要となる資源量を算出する。

【0034】

次に、算出された資源量を割当資源量として設定し（ステップ703）、その後の資源の割当状況を監視する（ステップ704）。そして、予測・設定された割当資源量と実際に使用された資源量とを比較して、比較結果を管理者に提示する。

【0035】

自律運用モードでは、監視運用モードとは異なり、割当資源量を管理者が変更しなくても、自動的に割当資源量が調整される。管理者は、割当資源量と使用資源量が運用システムとして適合していない場合には、自律運用モードから監視運用モードにモード変更を行うことができる。

【0036】

また、資源調整装置は、各モジュールの将来における長期的な資源量予測も行うことが可能である。例えば、1年間の使用資源量を2ヶ月間隔で予測した場合、図8のようなスケジュールが画面上に表示される。図8の曲線801、802、および803は、それぞれモジュール#1、#2、および#3の使用資源量のスケジュールを表す。

【0037】

システム内のすべてのモジュールについて予測された使用資源量を合計して、そのスケジュールをグラフ表示すれば、管理者に対してシステム全体の総需要を提示することができ、キャパシティプランニングの支援が実現される。

【0038】

次に、資源調整装置が行う処理についてより具体的に説明する。資源調整装置は、例えば、図9から図14までに示すようなテーブルを用いて資源調整処理を行う。

【0039】

図9のモジュール毎基本データテーブルには、モジュール毎に、以下の項目のデータが格納される。

モジュール名901（モジュール識別情報）

CPU割当量902

メモリ割当量903

ボトルネックCPU使用量検出閾値904

ボトルネックCPU使用量解消閾値905

ボトルネックメモリ使用量検出閾値906

ボトルネックメモリ使用量解消閾値907

ボトルネックトランザクション発生量検出閾値908

ボトルネックトランザクション発生量解消閾値909

シーリングポイント910（子プロセス生成指示値）

図9では、Module1およびModule2の基本データが示されている。

【0040】

図 1 0 および図 1 1 のモジュール毎統計データテーブルには、モジュール毎に、以下の項目のデータが格納される。

モジュール名 1 0 0 1 (モジュール識別情報)

取得間隔 1 0 0 2 (統計データを取得する時間間隔：可変)

取得時刻 1 0 0 3

トランザクション発生量 1 0 0 4

トランザクション処理量 1 0 0 5

C P U 使用量 1 0 0 6

C P U 割当量 1 0 0 7

C P U 予測使用量 1 0 0 8

メモリ使用量 1 0 0 9

メモリ割当量 1 0 1 0

メモリ予測使用量 1 0 1 1

このうち、1 0 0 3 ～ 1 0 1 1 の項目のデータは、取得時刻毎に繰り返し格納される。図 1 0 および図 1 1 では、Module 1 の統計データが 1 0 分毎に取得されており、2 0 0 3 年 3 月 1 日の 5 つの時刻のデータが例示されている。

【 0 0 4 1 】

図 1 2 および図 1 3 のモジュール毎サイクル情報データテーブルには、モジュール毎に、以下の項目のデータが格納される。

モジュール名 1 2 0 1 (モジュール識別情報)

サイクル期間 1 2 0 2

期間内間隔 1 2 0 3

時刻 1 2 0 4

トランザクション発生量 1 2 0 5

サイクル期間 1 3 0 1

期間内間隔 1 3 0 2

時刻 1 3 0 3

トランザクション発生量 1 3 0 4

このうち、1 2 0 2 ～ 1 2 0 5 の項目のデータは、サイクル期間 1 2 0 2 によ

り指定されるサイクル毎に繰り返し格納され、1204および1205の項目のデータは、さらに、期間内間隔1203により指定される時間間隔の時刻毎に繰り返し格納される。

【0042】

また、1301～1304の項目のデータは、複数のサイクルのデータを合成したデータを表し、サイクル期間1301としては、最も長いサイクル期間1202が設定され、期間内間隔1302としては、最も短いサイクル期間1202の期間内間隔1203が設定される。1303および1304の項目のデータは、期間内間隔1302により指定される時間間隔の時刻毎に繰り返し格納される。

【0043】

図12では、1日、1週間、および1ヶ月の各サイクルについて、期間内間隔の時刻毎にModule1のトランザクション発生量が示されており、図13では、1ヶ月分のトランザクション発生量の合成データが示されている。

【0044】

図14のモジュール毎資源予測データテーブルには、モジュール毎に、以下の項目のデータが格納される。

モジュール名1401（モジュール識別情報）

予測間隔1402（予測データの時間間隔：可変）

時刻1403

CPU使用量予測1404

トランザクション発生量予測1405

このうち、1403～1405の項目のデータは、予測間隔1402により指定される時間間隔の時刻毎に繰り返し格納される。図14では、Module1の資源予測データが1時間毎に生成されている。

【0045】

資源調整装置は、まず、管理者からの指示に従って、図9のモジュール毎基本データテーブルのデータを設定する。次に、図10および図11のモジュール毎統計データテーブル内に、モジュール毎のトランザクション量（発生量・処理量

）と、CPUおよびメモリの（使用量、割当量、予測使用量）を、取得間隔毎に蓄積する。以下の説明では、使用資源量としてCPU使用量に着目することにする。

【0046】

次に、図3に示したように、過去のトランザクション発生量をサイクル（日、週、月等）毎に表示し、さらに平均値と最大値を計算して表示する。管理者がサイクル毎に平均値もしくは最大値を選択すると、選択された値を図12のモジュール毎サイクル情報データテーブルに格納する。

【0047】

その後、例えば、“月”のトランザクション発生量の推移に“週”や“日”のトランザクション発生量の推移を加味して合成することで、図4のような“月”のトランザクション発生量の推移を算出し、そのデータを図13の合成データとして格納する。この合成データが、トランザクション発生量の予測データとなる。

【0048】

なお、管理者は、合成データの生成に使用するサイクルの種類を選択することが可能であり、資源調整装置は、選択されたサイクルのトランザクション発生量のみを用いて合成データを生成する。例えば、管理者が“日”のサイクルのみを選択して“月”のトランザクション発生量の推移の予測を指示した場合、資源調整装置は、“日”のトランザクション発生量の推移を繰り返しコピーする等の方法で、“月”のトランザクション発生量の推移を合成する。

【0049】

次に、モジュール毎統計データテーブル内の過去のトランザクション処理量とCPU使用量の関係から、図5および次式に示すような近似関数を求める。

$$(\text{CPU使用量}) = A * (\text{トランザクション処理量})^B + C$$

次に、モジュール毎サイクル情報データテーブルに格納されている“月”の合成データにおけるトランザクション発生量を、近似関数のトランザクション処理

量に代入して、“月”のCPU使用量の予測推移を計算する。

【0050】

そして、図6に示したように、今後の予測されるCPU使用量をスケジューリングし、図14のモジュール毎資源予測データテーブルにCPU使用量予測として格納する。また、予測されたトランザクション発生量は、トランザクション発生量予測としてモジュール毎資源予測データテーブルに格納される。その後、CPU使用量予測のデータは、モジュール毎統計データテーブルのCPU予測使用量の項目に過去データとして書き込まれる。

【0051】

また、モジュール毎統計データテーブルのCPU割当量は、CPU使用量予測のデータを用いて算出され、通常は、 $\text{CPU割当量} = \text{CPU予測使用量} (= \text{CPU使用量予測})$ となる。

【0052】

このような処理により、各サイクルを加味したモジュール毎の今後の使用資源量を的確に制限し、割当資源量を自動的に変動させることができる。したがって、システム全体の資源量を簡単に制御することが可能となる。

【0053】

資源調整装置により計算された使用資源量のスケジュールにおいて、管理者は、過去の経験等から使用資源量の推移を変更することで、今後の資源量の制限に関する設定を柔軟に変更することができる。このような変更操作を許すことで、過去のサイクルだけからは分からない資源量変動（例えば、企業の特別な日の使用資源量）に対しても、的確な制御を行うことができる。

【0054】

例えば、図6が“月”のCPU使用量のスケジュールを示している場合、管理者は折れ線グラフを変形することで、モジュール毎にCPU使用量の予測値を変更することができる。変更結果は、図14のモジュール毎資源予測データテーブルのCPU使用量予測として格納される。この場合、図10および図11のモジュール毎統計データテーブルの該当する取得時刻において、 $\text{CPU予測使用量} \neq \text{CPU割当量}$ となる（例えば、図10の2003/3/1/12:10のデータ

）。

【0055】

また、資源調整装置は、直近のトランザクション発生量の測定値から直後のトランザクション発生量の予測値を求め、その予測値と近似関数を用いて直後の使用資源量を予測してもよい。これにより、より確実な今後の資源量の制限を自動的に設定することが可能となる。

【0056】

上述の例では、今後のトランザクション発生量を予測しているが、過去のデータからでは計り知れない大量のトランザクションが突発的に発生することもある。そこで、資源調整装置は、直近（例えば、過去1時間）のトランザクション発生量を図10および図11のモジュール毎統計データテーブルから取得し、それを加味したトランザクション発生量の推移を生成する。

【0057】

そして、上述の近似関数を、直近のトランザクション発生量を加味したトランザクション発生量の推移に当てはめ、直後（例えば、今後1時間）のCPU使用量を予測して、図14のモジュール毎資源予測データテーブルのCPU使用量予測を書き換える。

【0058】

このような予測方法によれば、直近のトランザクション発生量からより現時点に近い使用資源量を予測できるので、突発的にトランザクション発生量が増大した場合でも、的確な制御を行うことができる。

【0059】

また、図9に示したように、モジュール毎に使用資源量割り当てのボトルネック検出閾値を設定しておけば、あるモジュールの使用資源量はその閾値に達したときに他のモジュールの資源を分け与えることができる。この処理は、使用資源量がボトルネック解消閾値に達するまで継続される。

【0060】

このように、使用資源量が一定値に達したモジュールに対して資源を優先的に割り当てることで、ボトルネックを回避することができる。これにより、例えば

、アプリケーションの極端なレスポンス遅延が防止され、システム全体の資源量が有効に活用される。

【0061】

一例として、図10および図11のモジュール毎統計データテーブルにおいて、Module 1の直前のCPU使用量が、図9のモジュール毎基本データテーブル内のボトルネックCPU使用量検出閾値（60%）に達した場合（2003/3/15:00）について説明する。この場合、資源調整装置は、図14のモジュール毎資源予測データテーブルを書き換えることにより、Module 1の今後のCPU使用量予測を大きく設定し（例えば、80%）、他のモジュールのCPU使用量予測を小さく設定する（例えば、20%）。

【0062】

その後、Module 1のCPU使用量が、モジュール毎基本データテーブル内のボトルネックCPU使用量解消閾値（40%）に達すると（2003/3/1/18:00）、モジュール毎資源予測データテーブルの各モジュールのCPU使用量予測を元に戻す。

【0063】

この例では、ボトルネックCPU使用量検出閾値およびボトルネックCPU使用量解消閾値を用いてボトルネック対策を行っているが、CPU使用量が増大しても、トランザクション発生量とトランザクション処理量がほぼ同じであれば、ボトルネック対策を行う必要はない。

【0064】

このように、あるモジュールが割当資源量を使い切っている場合でも、必ずしも資源が不足しているとは限らないので、使用資源量割り当てのボトルネック検出閾値を用いているだけでは、本当に他のモジュールの資源を分け与える必要があるかどうか分からない。

【0065】

そこで、ボトルネック検出閾値およびボトルネック解消閾値として、トランザクション発生量（例えば、処理待ち行列の長さ）を用いることが考えられる。例えば、処理待ち行列の長さを閾値として用いることで、本当に資源が不足してい

る状況を検出することができ、緊急時の資源割り当てが可能となる。これにより、モジュールの処理の停滞が発生するまでは、他のモジュールの資源を奪わずにシステム運用を続けることができ、システム全体の資源量が有効に活用される。

【0066】

一例として、図10および図11のモジュール毎統計データテーブルにおいて、Module1の直前のトランザクション発生量が、図9のモジュール毎基本データテーブル内のボトルネックトランザクション発生量検出閾値(500)に達した場合(2003/3/1/15:00)について説明する。この場合、資源調整装置は、図14のモジュール毎資源予測データテーブルを書き換えることにより、Module1の今後のCPU使用量予測を大きく設定し(例えば、80%)、他のモジュールのCPU使用量予測を小さく設定する(例えば、20%)。

【0067】

その後、Module1のトランザクション発生量が、モジュール毎基本データテーブル内のボトルネックトランザクション発生量解消閾値(300)に達すると(2003/3/1/18:00)、モジュール毎資源予測データテーブルの各モジュールのCPU使用量予測を元に戻す。

【0068】

モジュール毎基本データテーブル内のボトルネックCPU使用量検出/解消閾値とボトルネックトランザクション発生量検出/解消閾値を併用すると混乱を招くため、各モジュールの特性等を考慮して、いずれか一方を選択して使用することが望ましい。

【0069】

また、図8に示したように、資源調整装置は、全モジュールの長期的な(未来の)合計需要を予測し、予測結果をグラフィカルユーザインタフェース(GUI)の画面上に表示することで、管理者によるキャパシティプランニングを支援することができる。管理者は、提示された合計需要を参考にすることで、使用システムの構成変更等を適切に実行することが可能となる。

【0070】

この場合、資源調整装置は、図 1 0 および図 1 1 のモジュール毎統計データテーブル内のトランザクション発生量を長期間（例えば、数ヶ月から 1 年間）にわたって取得し、その期間のトランザクション処理量と資源量（例えば、CPU 使用量）の関係を表す近似関数を生成する。

【 0 0 7 1 】

次に、取得したトランザクション発生量の推移を予測推移として用い、この予測推移に近似関数を適用して、今後の長期的な（例えば、1 ～ 2 年間）使用資源量を予測する。そして、全モジュールの予測結果を合計することでシステム全体の合計需要を算出して、管理者に提示する。

【 0 0 7 2 】

このように、キャパシティプランニング支援の場合は、トランザクション発生量の合成データの代わりに、長期間にわたって取得されたトランザクション発生量のデータがそのものを用いて予測が行われる。

【 0 0 7 3 】

また、資源調整装置は、あるモジュールの使用資源量の予測値が上昇した場合に、そのモジュール内のアプリケーションに対して子プロセス生成等を促す通知を行うこともできる。アプリケーションに対してこのような通知を行うことで、そのアプリケーションは、処理待ちのトランザクション量を削減するために敏速な対応を行うことが可能となる。

【 0 0 7 4 】

アプリケーションの中には、使用資源量をいくら与えても、プロセスの数が少ないためにトランザクション処理量が増えないようなポイントを持つものがある。そこで、そのポイントの使用資源量をシーリングポイントとして、図 9 のモジュール毎基本データテーブルに設定しておく。

【 0 0 7 5 】

そして、図 1 4 のモジュール毎資源予測データテーブル内の該当モジュールの直近の CPU 使用量予測が、モジュール毎基本データテーブル内のシーリングポイントに達した場合、資源調整装置は、該当モジュールへ子プロセス生成等の指示を与える。指示を受けたモジュールのアプリケーションが子プロセスを生成す

れば、同時に処理できるトランザクションの数が増えるため、トランザクション処理量が増大する。

【0076】

このような現象が発生するアプリケーションが複数ある場合、それらのアプリケーションの集合を1つのモジュールとして管理すれば、アプリケーションへの通知を効率よく行うことができる。

【0077】

以上説明した実施形態では、主としてCPUを資源として扱っているが、本発明の資源調整方法は、メモリ、ディスク領域、データ転送帯域幅等を含む任意のコンピュータ資源に対して適用可能である。

【0078】

ところで、図1の資源調整装置は、例えば、図15に示すような情報処理装置（コンピュータ）を用いて構成することができる。図15の情報処理装置は、CPU（中央処理装置）1501、メモリ1502、入力装置1503、出力装置1504、外部記憶装置1505、媒体駆動装置1506、ネットワーク接続装置1507を備え、それらはバス1508により互いに接続されている。

【0079】

メモリ1502は、例えば、ROM（read only memory）、RAM（random access memory）等を含み、処理に用いられるプログラムおよびデータを格納する。CPU1501は、メモリ1502を利用してプログラムを実行することにより、必要な処理を行う。図9から図14までに示した各種のテーブルは、メモリ1502に格納されたデータに対応する。

【0080】

入力装置1503は、例えば、キーボード、ポインティングデバイス、タッチパネル等であり、オペレータからの指示や情報の入力に用いられる。出力装置1504は、例えば、ディスプレイ、プリンタ、スピーカ等であり、オペレータへの問い合わせや処理結果の出力に用いられる。

【0081】

外部記憶装置1505は、例えば、磁気ディスク装置、光ディスク装置、光磁

気ディスク装置、テープ装置等である。情報処理装置は、この外部記憶装置 1505 に、上記プログラムおよびデータを格納しておき、必要に応じて、それらをメモリ 1502 にロードして使用する。外部記憶装置 1505 は、図 9 から図 14 までに示した各種のテーブルを保存するためにも用いられる。

【0082】

媒体駆動装置 1506 は、可搬記録媒体 1509 を駆動し、その記録内容にアクセスする。可搬記録媒体 1509 は、メモリカード、フレキシブルディスク、CD-ROM (compact disk read only memory)、光ディスク、光磁気ディスク等の任意のコンピュータ読み取り可能な記録媒体である。オペレータは、この可搬記録媒体 1509 に上記プログラムおよびデータを格納しておき、必要に応じて、それらをメモリ 1502 にロードして使用する。

【0083】

ネットワーク接続装置 1507 は、LAN (local area network)、インターネット等の任意の通信ネットワークに接続され、通信に伴うデータ変換を行う。情報処理装置は、必要に応じて、上記プログラムおよびデータを外部の装置からネットワーク接続装置 1507 を介して受け取り、それらをメモリ 1502 にロードして使用する。

【0084】

図 16 は、図 15 の情報処理装置にプログラムおよびデータを供給することのできるコンピュータ読み取り可能な記録媒体を示している。可搬記録媒体 1509 やサーバ 1601 のデータベース 1603 に格納されたプログラムおよびデータは、情報処理装置 1602 のメモリ 1502 にロードされる。サーバ 1601 は、そのプログラムおよびデータを搬送する搬送信号を生成し、ネットワーク上の任意の伝送媒体を介して情報処理装置 1602 に送信する。CPU 1501 は、そのデータを用いてそのプログラムを実行し、必要な処理を行う。

【0085】

(付記 1) 1 つ以上のアプリケーションプログラムからなるモジュールを複数有するシステムにおいて使用されるコンピュータ資源の量をモジュール毎に調整する資源調整装置であって、

複数のモジュールのそれぞれについて、過去のトランザクション発生量の推移を表すデータを格納する格納手段と、

前記格納手段から対象モジュールのトランザクション発生量の推移を表すデータを取得し、該対象モジュールの過去のトランザクション処理量と使用資源量の相関関係を表す関数におけるトランザクション処理量としてトランザクション発生量を用いることで、該対象モジュールのトランザクション発生量の推移から使用資源量の推移を生成する生成手段と、

生成された使用資源量の推移を予測される使用資源量の推移として用い、該予測される使用資源量の推移に従って、前記対象モジュールの割当資源量を自動的に変動させる割当手段と
を備えることを特徴とする資源調整装置。

【 0 0 8 6 】

(付記 2) 1 つ以上のアプリケーションプログラムからなるモジュールを複数有するシステムにおいて使用されるコンピュータ資源の量をモジュール毎に調整するコンピュータのためのプログラムであって、

複数のモジュールのそれぞれについて過去のトランザクション発生量の推移を表すデータを格納する格納手段から、対象モジュールのトランザクション発生量の推移を表すデータを取得し、

前記対象モジュールの過去のトランザクション処理量と使用資源量の相関関係を表す関数におけるトランザクション処理量としてトランザクション発生量を用いることで、前記対象モジュールのトランザクション発生量の推移から使用資源量の推移を生成し、

生成された使用資源量の推移を予測される使用資源量の推移として用い、該予測される使用資源量の推移に従って、前記対象モジュールの割当資源量を自動的に変動させる
処理を前記コンピュータに実行させることを特徴とするプログラム。

【 0 0 8 7 】

(付記 3) 前記プログラムは、前記対象モジュールのトランザクション発生量の推移を表すデータから、複数種類のサイクルのそれぞれにおけるトランザク

ション発生量の推移を生成して画面上に表示し、オペレータからの指示に従って各サイクルにおけるトランザクション発生量の推移を合成することで、予測されるトランザクション発生量の推移を生成し、該予測されるトランザクション発生量の推移に前記関数を適用して前記使用資源量の推移を生成する処理を前記コンピュータに実行させることを特徴とする付記2記載のプログラム。

【0088】

(付記4) 前記プログラムは、前記複数種類のサイクルのそれぞれにおいて、前記システム内の2つ以上のモジュールに関するトランザクション発生量の平均値および最大値の推移を生成して前記画面上に表示し、前記オペレータが選択した値を用いて各サイクルにおけるトランザクション発生量の推移を合成することで、前記予測されるトランザクション発生量の推移を生成する処理を前記コンピュータに実行させることを特徴とする付記3記載のプログラム。

【0089】

(付記5) 前記プログラムは、前記生成された使用資源量の推移を画面上に表示し、オペレータが表示された使用資源量の推移を変更したとき、変更された使用資源量の推移を前記予測される使用資源量の推移として用いる処理を前記コンピュータに実行させることを特徴とする付記2記載のプログラム。

【0090】

(付記6) 前記プログラムは、前記格納手段から前記対象モジュールの直近のトランザクション発生量の推移を表すデータを取得し、該直近のトランザクション発生量の推移から生成された使用資源量の推移を直後の予測される使用資源量の推移として用いて、該対象モジュールの直後の割当資源量を変動させる処理を前記コンピュータに実行させることを特徴とする付記2記載のプログラム。

【0091】

(付記7) 前記プログラムは、前記対象モジュールの使用資源量があらかじめ決められたボトルネック検出閾値に達したときから、該対象モジュールの使用資源量があらかじめ決められたボトルネック解消閾値に達するまでの間、該対象モジュールに優先的に資源を割り当てる処理を前記コンピュータに実行させることを特徴とする付記2記載のプログラム。

【0092】

(付記8) 前記プログラムは、前記対象モジュールのトランザクション発生量があらかじめ決められたボトルネック検出閾値に達したときから、該対象モジュールのトランザクション発生量があらかじめ決められたボトルネック解消閾値に達するまでの間、該対象モジュールに優先的に資源を割り当てる処理を前記コンピュータに実行させることを特徴とする付記2記載のプログラム。

【0093】

(付記9) 前記プログラムは、前記対象モジュールの予測される使用資源量があらかじめ決められた値に達したとき、該対象モジュールに対して子プロセスの生成を指示する処理を前記コンピュータに実行させることを特徴とする付記2記載のプログラム。

【0094】

(付記10) 前記プログラムは、長期間にわたって予測される使用資源量の推移を含むキャパシティプランニング支援のための画面を表示する処理を前記コンピュータに実行させることを特徴とする付記2記載のプログラム。

【0095】

(付記11) 1つ以上のアプリケーションプログラムからなるモジュールを複数有するシステムにおいて使用されるコンピュータ資源の量をモジュール毎に調整するコンピュータのためのプログラムを記録した記録媒体であって、該プログラムは、

複数のモジュールのそれぞれについて過去のトランザクション発生量の推移を表すデータを格納する格納手段から、対象モジュールのトランザクション発生量の推移を表すデータを取得し、

前記対象モジュールの過去のトランザクション処理量と使用資源量の相関関係を表す関数におけるトランザクション処理量としてトランザクション発生量を用いることで、前記対象モジュールのトランザクション発生量の推移から使用資源量の推移を生成し、

生成された使用資源量の推移を予測される使用資源量の推移として用い、該予測される使用資源量の推移に従って、前記対象モジュールの割当資源量を自動的

に変動させる

処理を前記コンピュータに実行させることを特徴とするコンピュータ読み取り可能な記録媒体。

【0096】

(付記12) 1つ以上のアプリケーションプログラムからなるモジュールを複数有するシステムにおいて使用されるコンピュータ資源の量をモジュール毎に調整するコンピュータのためのプログラムを搬送する搬送信号であって、該プログラムは、

複数のモジュールのそれぞれについて過去のトランザクション発生量の推移を表すデータを格納する格納手段から、対象モジュールのトランザクション発生量の推移を表すデータを取得し、

前記対象モジュールの過去のトランザクション処理量と使用資源量の相関関係を表す関数におけるトランザクション処理量としてトランザクション発生量を用いることで、前記対象モジュールのトランザクション発生量の推移から使用資源量の推移を生成し、

生成された使用資源量の推移を予測される使用資源量の推移として用い、該予測される使用資源量の推移に従って、前記対象モジュールの割当資源量を自動的に変動させる

処理を前記コンピュータに実行させることを特徴とする搬送信号。

【0097】

(付記13) 1つ以上のアプリケーションプログラムからなるモジュールを複数有するシステムにおいて使用されるコンピュータ資源の量をモジュール毎に調整する資源調整方法であって、

生成手段が、複数のモジュールのそれぞれについて過去のトランザクション発生量の推移を表すデータを格納する格納手段から、対象モジュールのトランザクション発生量の推移を表すデータを取得し、

前記生成手段が、前記対象モジュールの過去のトランザクション処理量と使用資源量の相関関係を表す関数におけるトランザクション処理量としてトランザクション発生量を用いることで、前記対象モジュールのトランザクション発生量の

推移から使用資源量の推移を生成し、

割当手段が、生成された使用資源量の推移を予測される使用資源量の推移として用い、該予測される使用資源量の推移に従って、前記対象モジュールの割当資源量を自動的に変動させる

ことを特徴とする資源調整方法。

【0098】

【発明の効果】

本発明によれば、トランザクション発生量の測定値から今後の使用資源量を予測することで、モジュール毎の使用資源量の制限を自律的かつ的確に変動させることができる。

【図面の簡単な説明】

【図1】

本発明の資源調整装置の原理図である。

【図2】

監視運用モードの処理のフローチャートである。

【図3】

過去のトランザクション発生量の推移を示す図である。

【図4】

今後のトランザクション発生量の推移を示す図である。

【図5】

近似関数を示す図である。

【図6】

使用資源量のスケジュールを示す図である。

【図7】

自律運用モードの処理のフローチャートである。

【図8】

キャパシティプランニング支援を示す図である。

【図9】

モジュール毎基本データテーブルを示す図である。

【図 10】

モジュール毎統計データテーブルを示す図（その 1）である。

【図 11】

モジュール毎統計データテーブルを示す図（その 2）である。

【図 12】

モジュール毎サイクル情報データテーブルを示す図（その 1）である。

【図 13】

モジュール毎サイクル情報データテーブルを示す図（その 2）である。

【図 14】

モジュール毎資源予測データテーブルを示す図である。

【図 15】

情報処理装置の構成図である。

【図 16】

記録媒体を示す図である。

【図 17】

モジュールを示す図である。

【符号の説明】

101 格納手段

102 生成手段

103 割当手段

301、302、303、304、305、601、602、603 折れ線

グラフ

306、604 OKボタン

801、802、803 曲線

901、1001、1201、1401 モジュール名

902、1007 CPU割当量

903、1010 メモリ割当量

904 ボトルネックCPU使用量検出閾値

905 ボトルネックCPU使用量解消閾値

906 ボトルネックメモリ使用量検出閾値
907 ボトルネックメモリ使用量解消閾値
908 ボトルネックトランザクション発生量検出閾値
909 ボトルネックトランザクション発生量解消閾値
910 シーリングポイント
1002 取得間隔
1003 取得時刻
1004、1205、1304 トランザクション発生量
1005 トランザクション処理量
1006 CPU使用量
1008 CPU予測使用量
1009 メモリ使用量
1011 メモリ予測使用量
1202、1301 サイクル期間
1203、1302 期間内間隔
1204、1303、1403 時刻
1402 予測間隔
1404 CPU使用量予測
1405 トランザクション発生量予測
1501 CPU
1502 メモリ
1503 入力装置
1504 出力装置
1505 外部記憶装置
1506 媒体駆動装置
1507 ネットワーク接続装置
1508 バス
1509 可搬記録媒体
1601 サーバ

1 6 0 2 情報処理装置

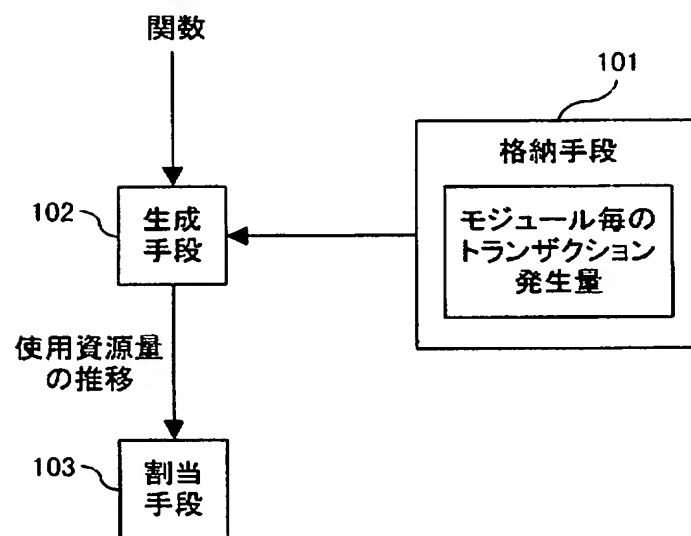
1 6 0 3 データベース

【書類名】

図面

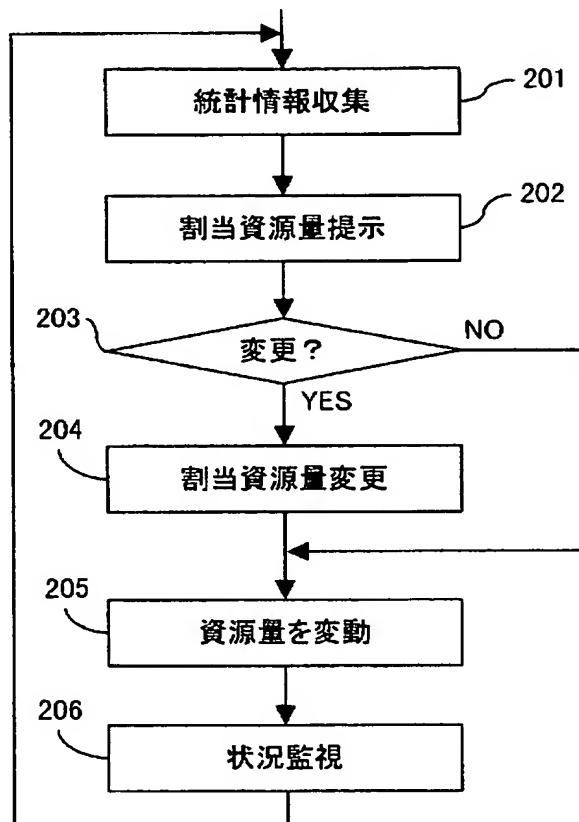
【図 1】

本発明の資源調整装置の原理図



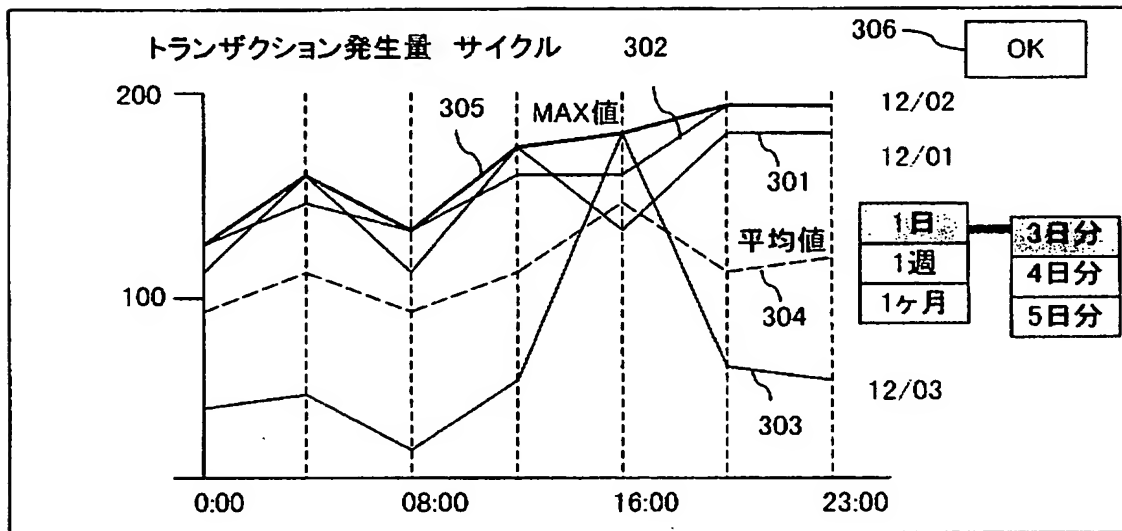
【図 2】

監視運用モードの処理のフローチャート



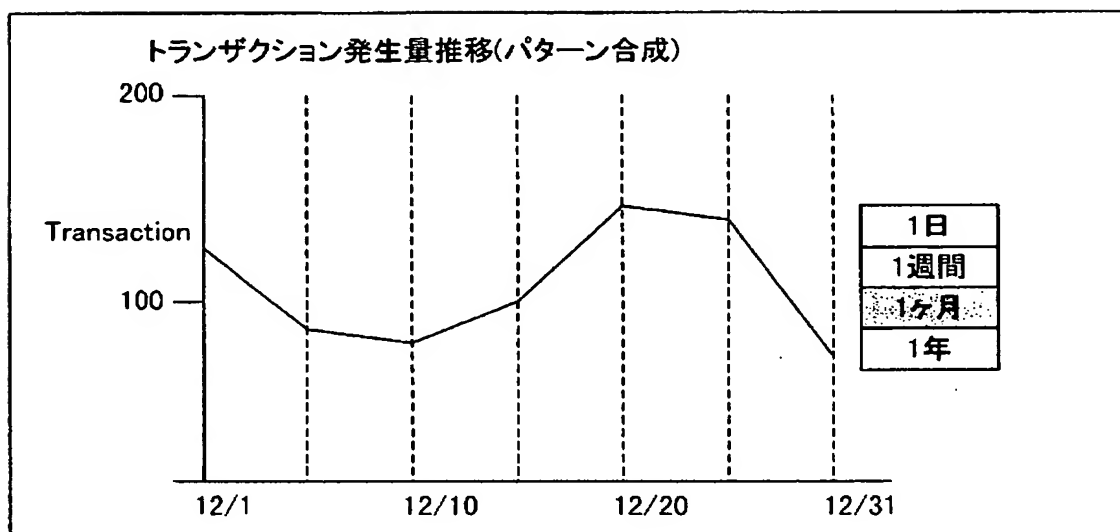
【図 3】

過去のトランザクション発生量の推移を示す図



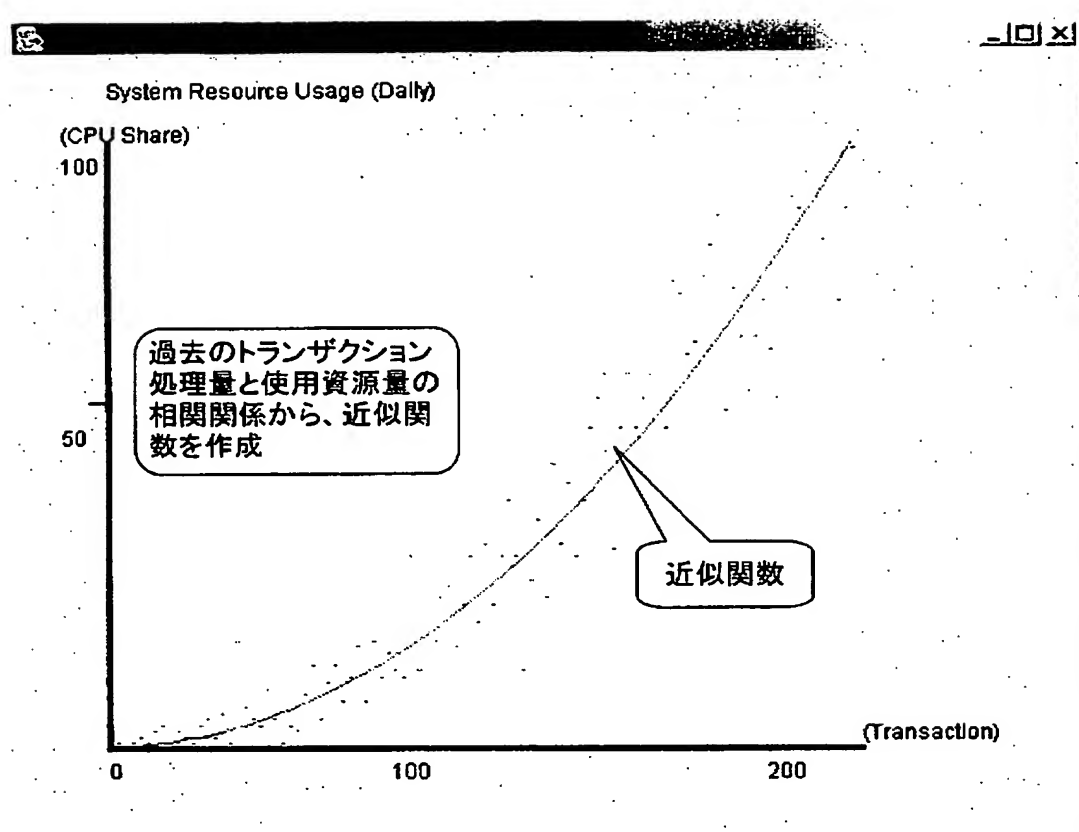
【図 4】

今後のトランザクション発生量の推移を示す図



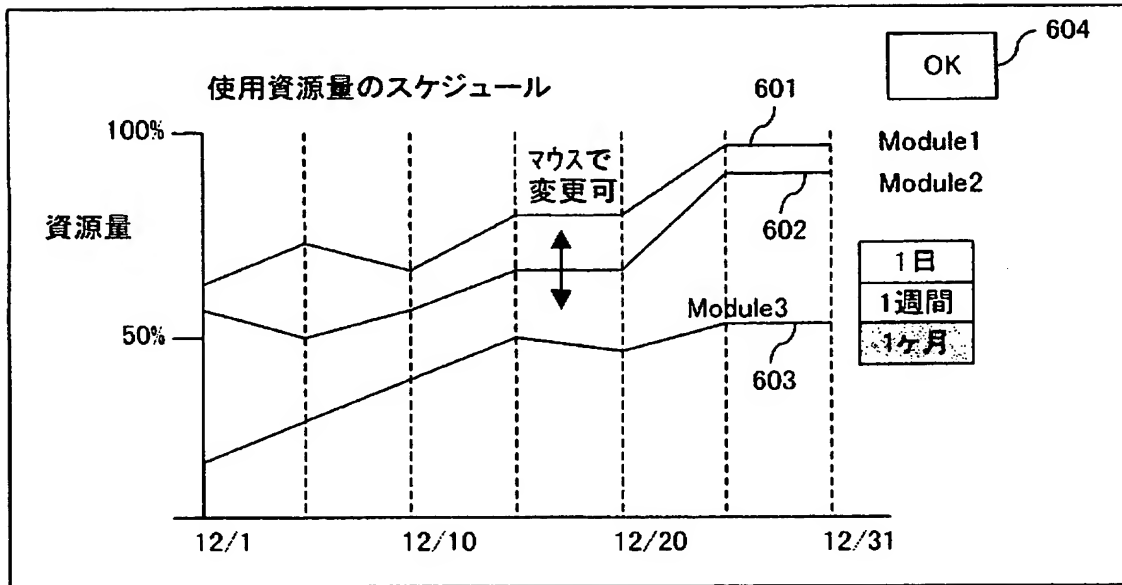
【図 5】

近似関数を示す図



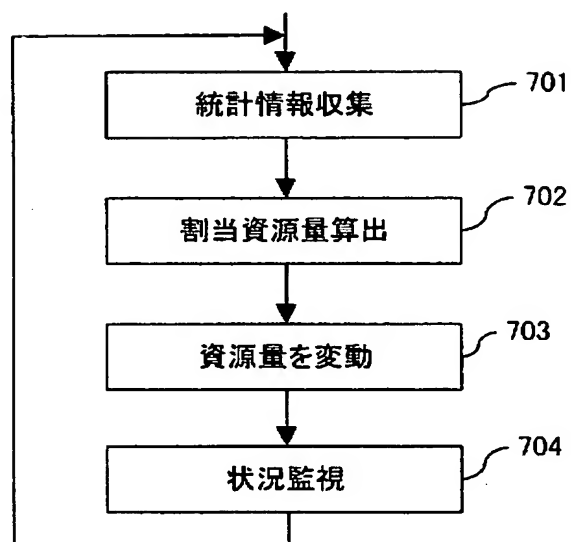
【図 6】

使用資源量のスケジュールを示す図



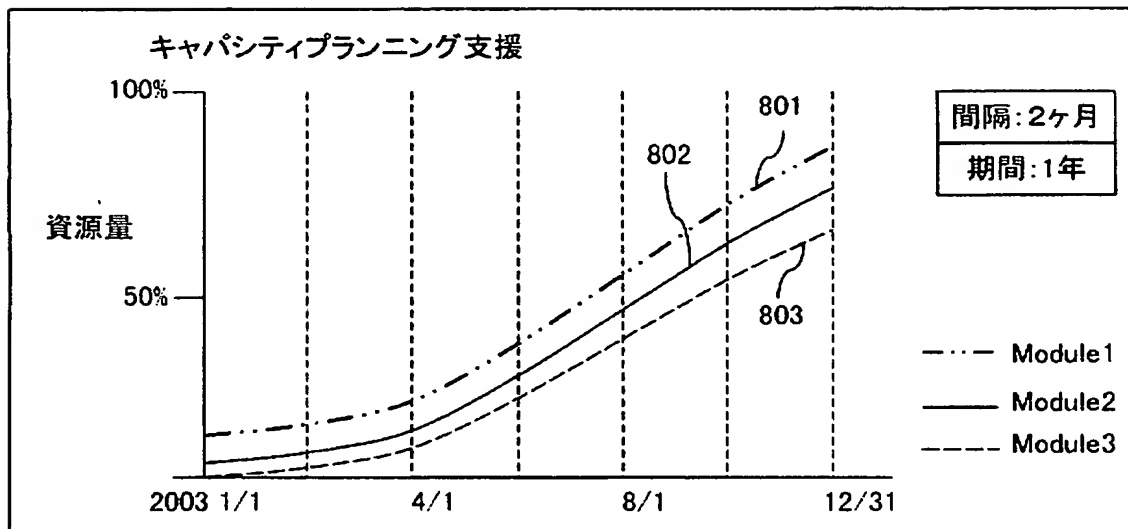
【図 7】

自律運用モードの処理のフローチャート



【図 8】

キャパシティプランニング支援を示す図



【図 9】

モジュール毎基本データテーブルを示す図

901	モジュール名	Module1
902	CPU割当量	50%
903	メモリ割当量	50%
904	ボトルネックCPU使用量検出閾値	60%
905	ボトルネックCPU使用量解消閾値	40%
906	ボトルネックメモリ使用量検出閾値	50%
907	ボトルネックメモリ使用量解消閾値	35%
908	ボトルネックトランザクション発生量検出閾値	500
909	ボトルネックトランザクション発生量解消閾値	300
910	シーリングポイント(子プロセス生成指示値)	70%
901	モジュール名	Module2
902	CPU割当量	50%
903	メモリ割当量	50%
904	ボトルネックCPU使用量検出閾値	50%
905	ボトルネックCPU使用量解消閾値	25%
906	ボトルネックメモリ使用量検出閾値	50%
907	ボトルネックメモリ使用量解消閾値	30%
908	ボトルネックトランザクション発生量検出閾値	300
909	ボトルネックトランザクション発生量解消閾値	100
910	シーリングポイント(子プロセス生成指示値)	60%

【図 10】

モジュール毎統計データテーブルを示す図(その1)

1001	モジュール名	Module1
1002	取得間隔	10分
1003	取得時刻	2003/3/1/12:00
1004	トランザクション発生量	200
1005	トランザクション処理量	100
1006	CPU使用量	20%
1007	CPU割当量	30%
1008	CPU予測使用量	30%
1009	メモリ使用量	30%
1010	メモリ割当量	25%
1011	メモリ予測使用量	20%
1003	取得時刻	2003/3/1/12:10
1004	トランザクション発生量	210
1005	トランザクション処理量	120
1006	CPU使用量	25%
1007	CPU割当量	20%
1008	CPU予測使用量	22%
1009	メモリ使用量	32%
1010	メモリ割当量	30%
1011	メモリ予測使用量	35%
	略	

【図 11】

モジュール毎統計データテーブルを示す図(その2)

1003	取得時刻	2003/3/1/15:00
1004	トランザクション発生量	500
1005	トランザクション処理量	120
1006	CPU使用量	60%
1007	CPU割当量	50%
1008	CPU予測使用量	50%
1009	メモリ使用量	50%
1010	メモリ割当量	35%
1011	メモリ予測使用量	40%
1003	略	
1004		
1005	取得時刻	2003/3/1/18:00
1006	トランザクション発生量	300
1007	トランザクション処理量	250
1008	CPU使用量	40%
1009	CPU割当量	80%
1010	CPU予測使用量	80%
1011	メモリ使用量	30%
	メモリ割当量	35%
	メモリ予測使用量	30%
	略	
1003	取得時刻	2003/3/1/23:50
1004	トランザクション発生量	200
1005	トランザクション処理量	180
1006	CPU使用量	30%
1007	CPU割当量	30%
1008	CPU予測使用量	25%
1009	メモリ使用量	20%
1010	メモリ割当量	25%
1011	メモリ予測使用量	20%
	以下略	

【図 12】

モジュール毎サイクル情報データテーブルを示す図(その1)

1201	モジュール名	Module1
1202	サイクル期間	1日
1203	期間内間隔	1時間
1204	時刻	00:00
1205	トランザクション発生量	150
1204	時刻	01:00
1205	トランザクション発生量	120
	略	
1204	時刻	23:00
1205	トランザクション発生量	180
1202	サイクル期間	1週間
1203	期間内間隔	1日
1204	時刻	日曜日
1205	トランザクション発生量	200
1204	時刻	月曜日
1205	トランザクション発生量	180
	略	
1204	時刻	土曜日
1205	トランザクション発生量	150
1202	サイクル期間	1ヶ月
1203	期間内間隔	1日
1204	時刻	1日
1205	トランザクション発生量	120
1204	時刻	2日
1205	トランザクション発生量	180
	略	
1204	時刻	31日
1205	トランザクション発生量	150

【図 13】

モジュール毎サイクル情報データテーブルを示す図(その2)

	合成データ	
1301	サイクル期間	1ヶ月
1302	期間内間隔	1時間
1303	時刻	1日0:00
1304	トランザクション発生量	180
1303	時刻	1日1:00
1304	トランザクション発生量	170
	略	
1303	時刻	31日 23:00
1304	トランザクション発生量	120

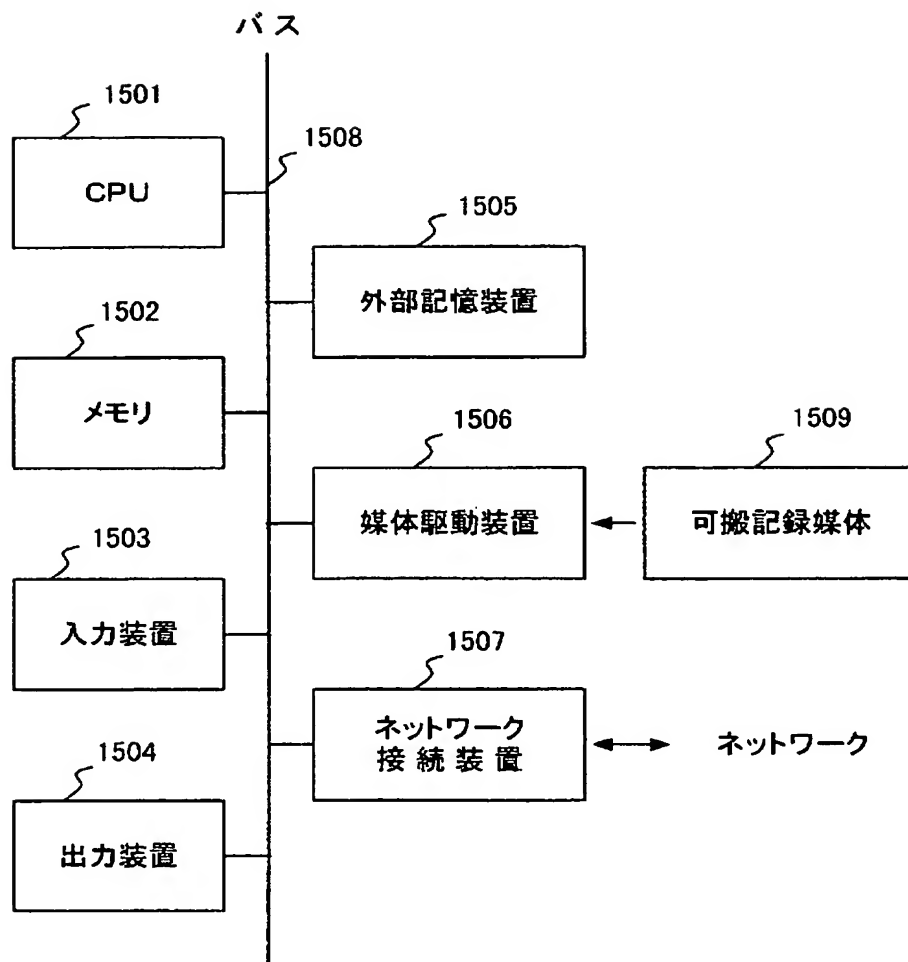
【図 14】

モジュール毎資源予測データテーブルを示す図

1401	モジュール名	Module1
1402	予測間隔	1時間
1403	時刻	2003/3/1 00:00
1404	CPU使用量予測	45%
1405	トランザクション発生量予測	200
1403	時刻	2003/3/1 01:00
1404	CPU使用量予測	43%
1405	トランザクション発生量予測	200
	略	
1403	時刻	2003/3/12 03:00
1404	CPU使用量予測	60%
1405	トランザクション発生量予測	500
	略	
1403	時刻	2003/3/12 01:00
1404	CPU使用量予測	40%
1405	トランザクション発生量予測	300
	略	
1403	時刻	2003/3/31 23:00
1404	CPU使用量予測	45%
1405	トランザクション発生量予測	200

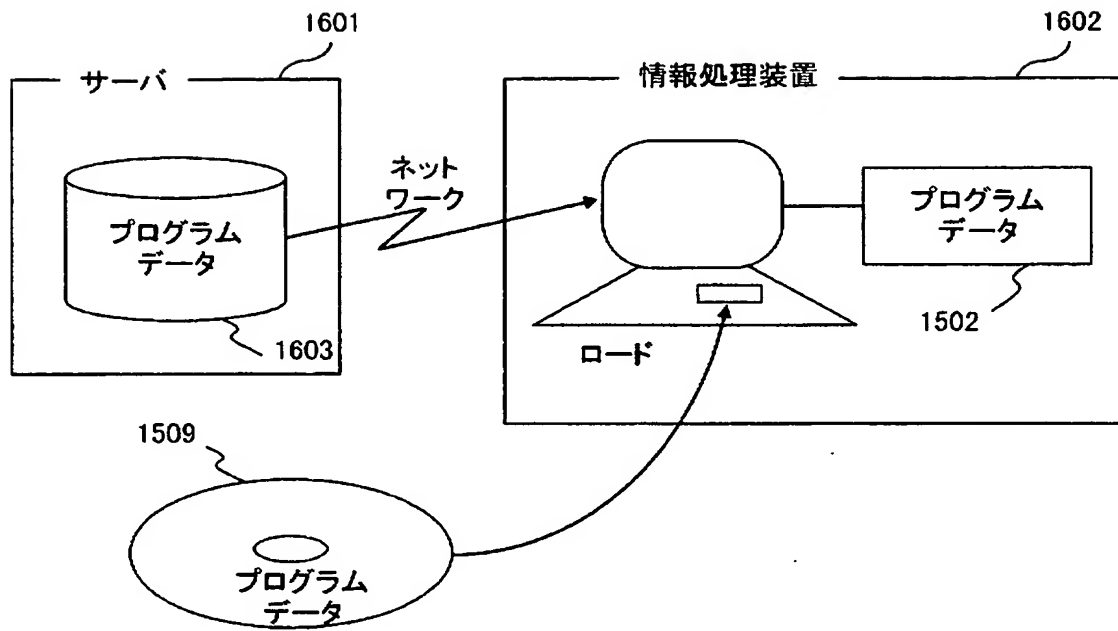
【図 15】

情報処理装置の構成図



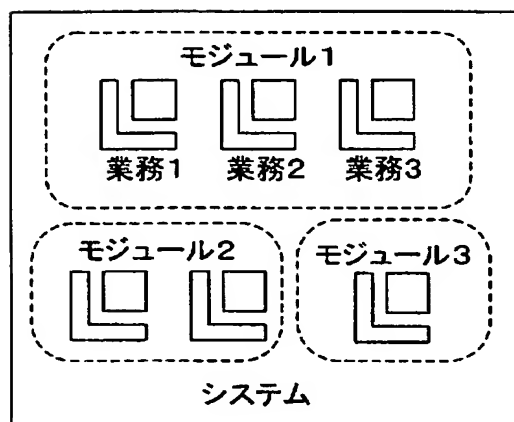
【図 16】

記録媒体を示す図



【図 17】

モジュールを示す図



【書類名】 要約書

【要約】

【課題】 情報処理システムにおける各モジュールの使用資源量の変化を的確に予測し、その変化に応じて使用資源量の制限を自律的に調整する。

【解決手段】 生成手段 1 0 2 は、格納手段 1 0 1 からモジュール毎のトランザクション発生量を取得し、過去のトランザクション処理量と使用資源量の相関関係を表す関数を用いて、トランザクション発生量の推移から予測される使用資源量の推移を生成する。割当手段 1 0 3 は、その使用資源量の推移に従って、モジュール毎の割当資源量を自動的に変動させる。

【選択図】 図 1



特願 2 0 0 3 - 1 1 1 3 0 1

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 5 2 2 3]

1. 変更年月日

1 9 9 6 年 3 月 2 6 日

[変更理由]

住所変更

住 所

神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号

氏 名

富士通株式会社